Spring 2016

Moodle Java Autograder

Prathibha Davuluri
Governors State University

Pranavi Reddy Madadi
Governors State University

ABSTRACT

The project is developing a “JAVA Automatic Grader” for java project assignment. It ensures that every assignment or work given to the students is done in the right way. Students should get automatic evaluation after submitting their assignment. We are implementing each test case in such a way that student should only get marks for satisfying all the requirements of the project assignment. We are also providing a direct tab for google users for easy access where one click directly takes your google ID and password to directly open Moodle account without typing in your user ID and password.

Table of Contents

1 Project Description

The project is developing a “JAVA Automatic Grader” for java project assignment. It ensures that every assignment or work given to the students is done in the right way. Students should get automatic evaluation after submitting their assignment. We are implementing each test case in such a way that student should only get marks for satisfying all the requirements of the project assignment. With the increasing enrollments in CS courses and the growing interest in providing MOOCs, automated grading of student programs is becoming a necessity. This paper describes our first experience of using Moodle's Virtual Programming Lab (VPL) for the automatic evaluation of students' program in two of our programming courses. Early experimentation in several courses show the module to be flexible and robust, allowing for sophisticated ways to test student programs. The automatic grading requires a significant shift of faculty time, putting the bulk of the effort up-front, preparing not only the assignment, but also the solution program, the testing environment, and its validation. However, once this phase is over, very little attention is needed. We are also providing a direct tab for google users for easy access where a click directly takes yours google ID and password to directly open Moodle account without typing in your user ID and password.

.

1.1 Business Requirement Information

As we know that we have a lot of websites which are used in universities to give an assignment or submit an assignment, and then the professors have to download students work and grade their work accordingly by checking each students work. Its hectic for the professors as they are busy with lot of other work. To make every ones’ work easier we have taken this project and in this the students work will be graded as soon as they submit their work online.

2 Project Technical Description

Furthermore, the richness of options VPL provides, its overall robustness, and its ability to incorporate the instructor's code as part of the test frame, have contributed to a successful experience at our institution, and growing expertise, and we plan on adopting the module more widely in the future. There are many caveats, though. The most noticeable is a time shift, requiring a significant amount of work before leasing an assignment, and practically none after the assignment is released. The instructor's time must now incorporate the preparation of the assignment, the preparation of the solution program, the creation of scripts that will evaluate the student program, the testing of the setup, and the creation of a new VPL activity on Moodle. Once these actions are performed, the instructor's attention to the assignment is minimal. After the assignment is closed, a brief review of the submissions will ensure that everybody received a grade, and a quick scan of submitted programs will help detect fraudulent attempts (such a writing a program that prints the solution instead of computing it). The VPL module provides a similarity test to flag submitted programs that have a high level of similitude, a useful feature for spotting possible fraud. After this half hour intervention, the instructor is basically free of the assignment.

2.1 Application Architecture

VPL is one of many Moodle modules. It requires a typicaltwo-step installation process to integrate with this Learning Management System. Although it doesn't require it, a dedicated separate execution server, or jail server for short, is highly recommended. This jail server runs the test scripts along with the programs submitted by the students. Should a student program crash the jail server, the Moodle server is unaffected? We provide a quick summary of the VPL operations now.

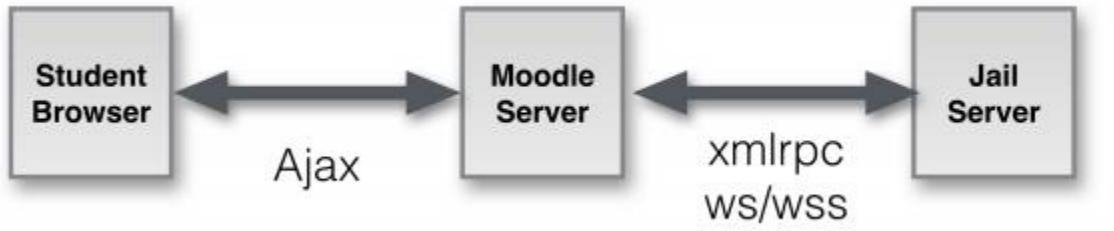

Figure 1: Technologies used in Student to Moodle VPL connections

3 Project Requirements

3.1 Identification of Requirements

For this project, few soft wares are to be downloaded below I am explaining all the requirements step by step.
All are interlinked, to install Moodle we need WAMP server. For the installation of WAMP server we need Visual C++ (Visual Studio). So to start with,

a. **Install Visual Studio 2012 or recent version**

Here we are installing basic visual studio 2012 of 64 bit (https://www.microsoft.com/en-us/download/confirmation.aspx?id=30682), as we need Visual C++. Installation is simple, but before that create a folder and place all downloads in this folder.

b. Install WAMP server

We need Apache, My SQL and PHP for Moodle installation we can download them separately but we preferred the combo package WAMP (Windows Apache MySQL PHP) http://www.wampserver.com/en/ . Download WAMP server 64bit in the same folder and follow the steps for installation. There will be letter 'W' on the task bar where initially it is in red color and then turns orange and then green. Green indicates that WAMP server is online. Now open Internet Explorer or Google Chrome and on the search bar type localhost if the WAMP server is opens, it means that you have installed correctly. Under projects you find add project as you haven't download Moodle software yet.

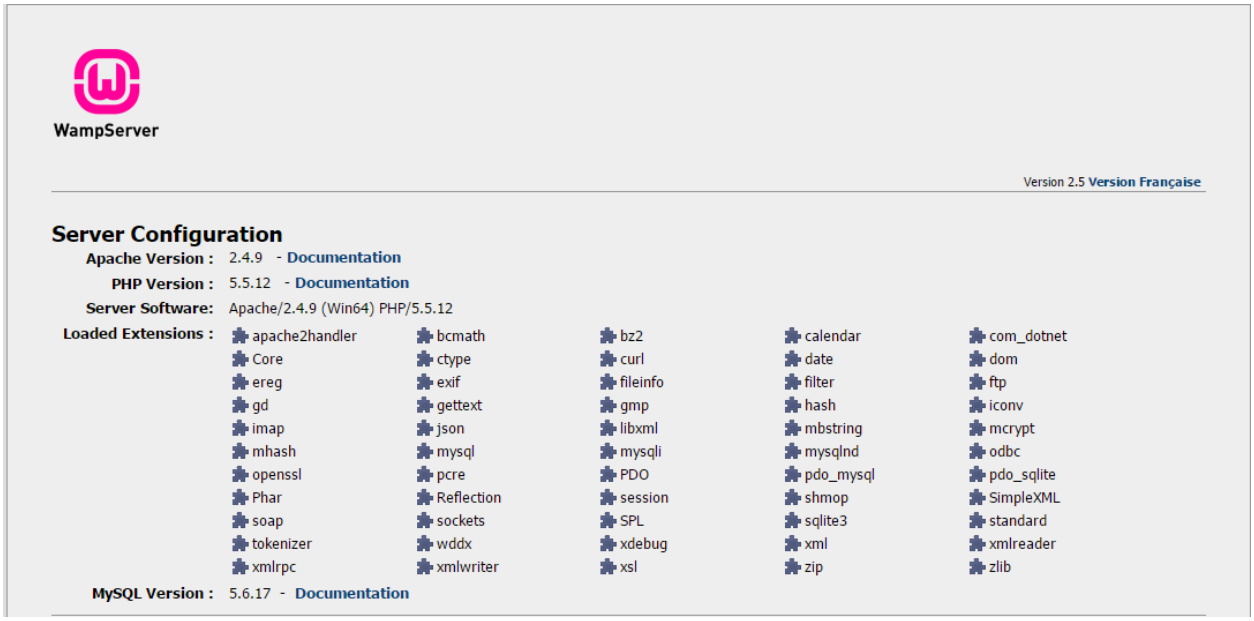

Figure 2: WAMP Server after installation

c. Install Moodle

Now it's time to download Moodle (Modular Object Oriented Dynamic Learning Environment) https://download.moodle.org/ software and download the 64 bit zip file in the same folder. Extract the folder in the "www" folder where it has been created while installation of WAMP server (C drive- wamp folder- www folder). Now again open Google Chrome and type local host you will now find Moodle under Projects. Click that Moodle option you will find installation page for Moodle choose language and follow the steps accordingly, this takes for about 1 hour minimum. Now after installation we need to name our page and we named it as "E-LEARNING".

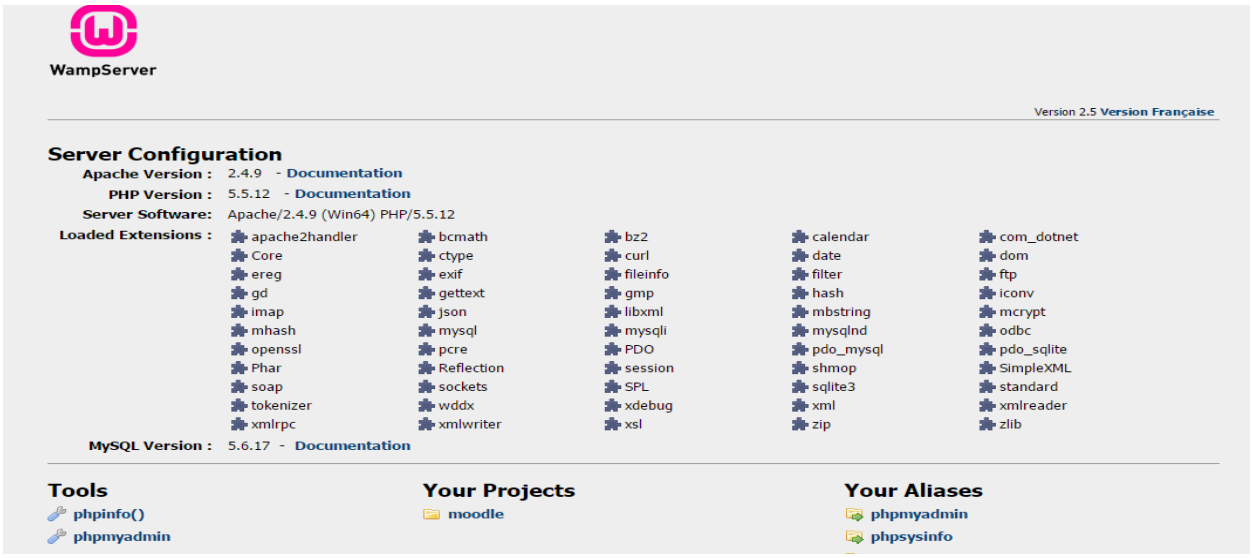

Figure 3: Wamp Server after downloading Moodle

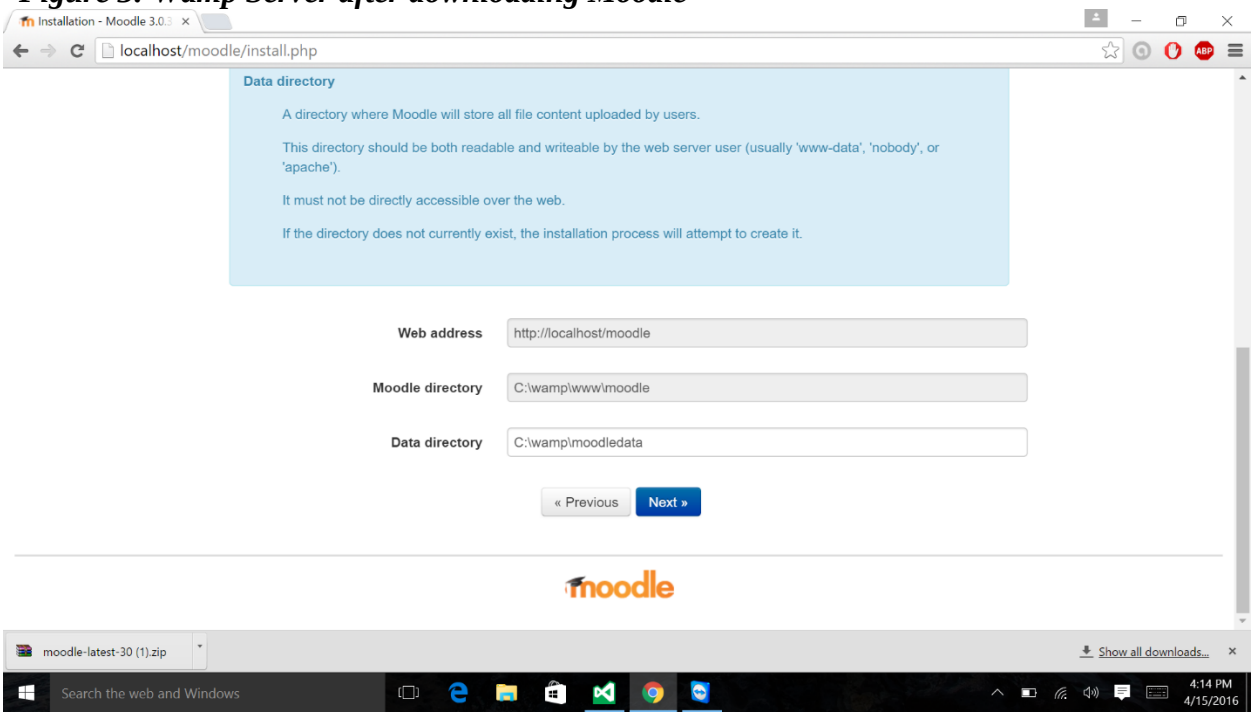

Figure 4: Moodle Installation, selection of directories

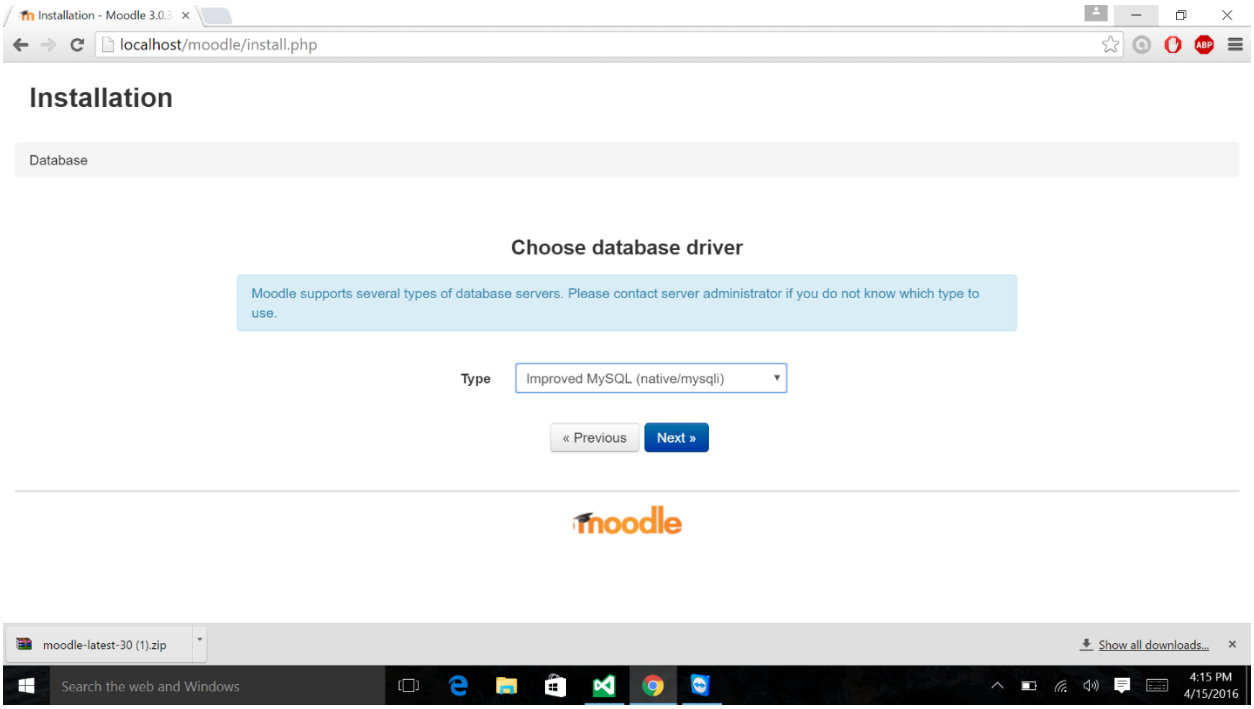

Figure 5: Moodle installation, selection of database driver.

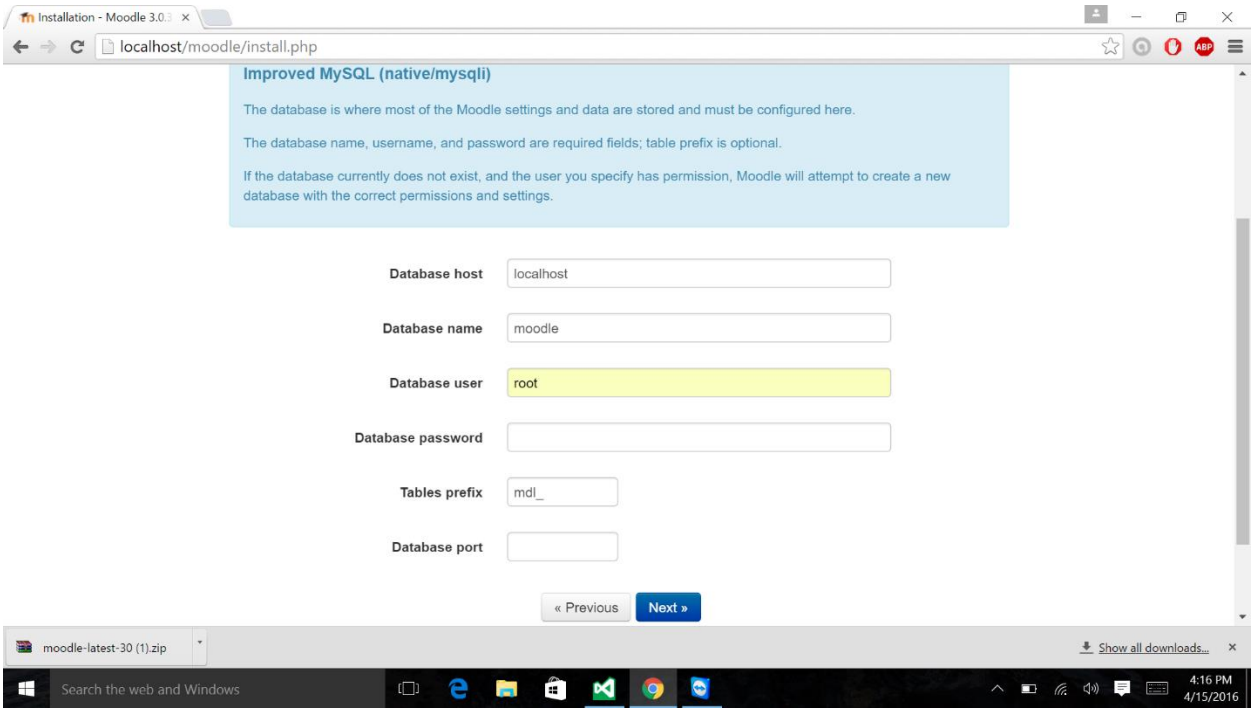

Figure 6: Moodle installation, Database user as root.

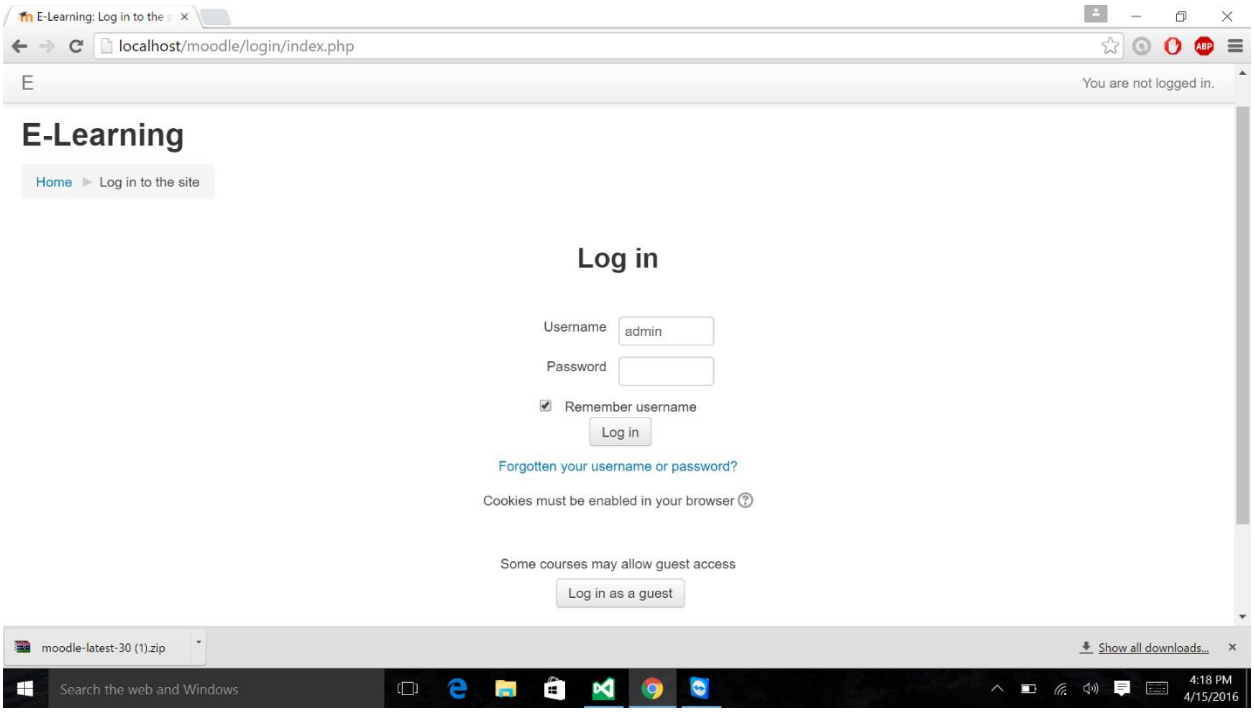

Figure 7: Moodle after installation "E-learning login page

3.2 Operations, Administration, Maintenance and Provisioning (OAM&P)

A student interacts with the system as illustrated in Figure 1. The students interact with VPL through a browser. When a student submits her program (duplicate/paste, edit, upload), the Moodle server packages the instructor's test script along with the student program in an archive, and ships it to the VPL jail server. There, the test scripts are executed in a sandbox environment, and the captured output is sent back to the Moodle server. The grade is automatically incorporated in the student's internal record, under the rubric selected by the instructor (lab, homework, quiz, etc.). Note that VPL has an automated feature that allows the instructor to define the input that must be fed to the student program, and the expected output for each input. VPL automatically assigns a grade between 0 and 100% proportional to the number of outputs that correctly match the expected outputs. This feature can be used for very simple assignments.

4 Project Design Description

The project design is simple, as the complete project is based on the Moodle and after we have downloaded the Moodle we are supposed to upload few plugins in for Moodle, Virtual Programming Language and we need the Google Sign in link we need to download and upload the Oauth2 Plugin.

a. Virtual Programming Lab (VPL).

We need VPL as we are doing a java auto-grader. VPL has plagiarism tool installed in it already, so it's a good package. Open Google Chrome and type VPL plugin for Moodle download or paste this link https://moodle.org/plugins/mod_vpl and download the zip file. To install this

plugin in Moodle, first login as admin, left corner you find under administration, site administration, plugins, install plugins, now select the zipped file and drop in the file block, click continue. VPL is now installed.

VPL sports many features. We list here those options we believe computer science instructors will find most interesting.

• The number of languages supported is quite large. The VPL Web site [1] lists Ada, C, C++, C#, Fortran, Haskell, Java, MatLab, Octave, Pascal, Perl, Php, Prolog, Python, Ruby, Scheme, SQL, and VHD languages as supported. We have successfully implemented VPL assignments for Python, Java, and assembly language. Any language with a compiler or interpreter supported by Linux with executable that output text can be evaluated. Testing programs outputting graphics requires additional tools and expertise.

• The instructor defines how the student program is evaluated and graded. This allows for testing properties of a program other than its output to a given input. For example, in assembly language, it may be important for an assignment to generate a program with as small a footprint in memory as possible. The instructor can create a script that will measure the static footprint of the student program and assign a grade inversely proportional to the program's byte size.

• VPL uses HTML5, Ajax, and WebSocket, in an effort to free students from having to us Java-enabled browsers.

• The instructor can define the rubric under which a VPL grade is assigned. This is controlled per VPL-task.

• The instructor can make the grade visible to the student, or not, and reveal it only after the due date.

• The instructor can limit the number of submissions for a given program, or set of programs. A survey of some 30 VPL assignments given this semester indicates that approximately 85% of the students submit a given program less than 10 times, while the remaining 15% fall in a long tail of recorded submission clicks. The largest recorded number of submission for a given assignment is 51 times.

• The instructor controls the resources needed by the jail server, including stack or dynamic space.

• For a given VPL activity, the programs submitted can be those of an individual student, or from a group of students.

• Access to the submission page can be restricted by IP address, a feature probably more useful in a MOOC environment, or protected by a password.

b. Oauth2.

Oauth2 is used to create direct links for easy Google, Facebook and Windows. Same procedure as VPL download Oauth2 plugin for Moodle download or paste this link https://moodle.org/plugins/auth_googleoauth2 and download zip file. To install this plugin in Moodle, first login as admin, left corner you find under administration, site administration, plugins, install plugins, now select the zipped file and drop in the file block, click continue, Oauth2 is installed. Now open Plugins overview, select additional plugins then Oauth2 enable it and enable blocks editing on, select settings, we need Google sign in so follow the steps stated in the settings beside google client. Paste Google Client ID and Google Client Secret from Google console API. Now refresh the E-Learning and logout. Now you can see that the sign-in with google option is created in the Login Page.

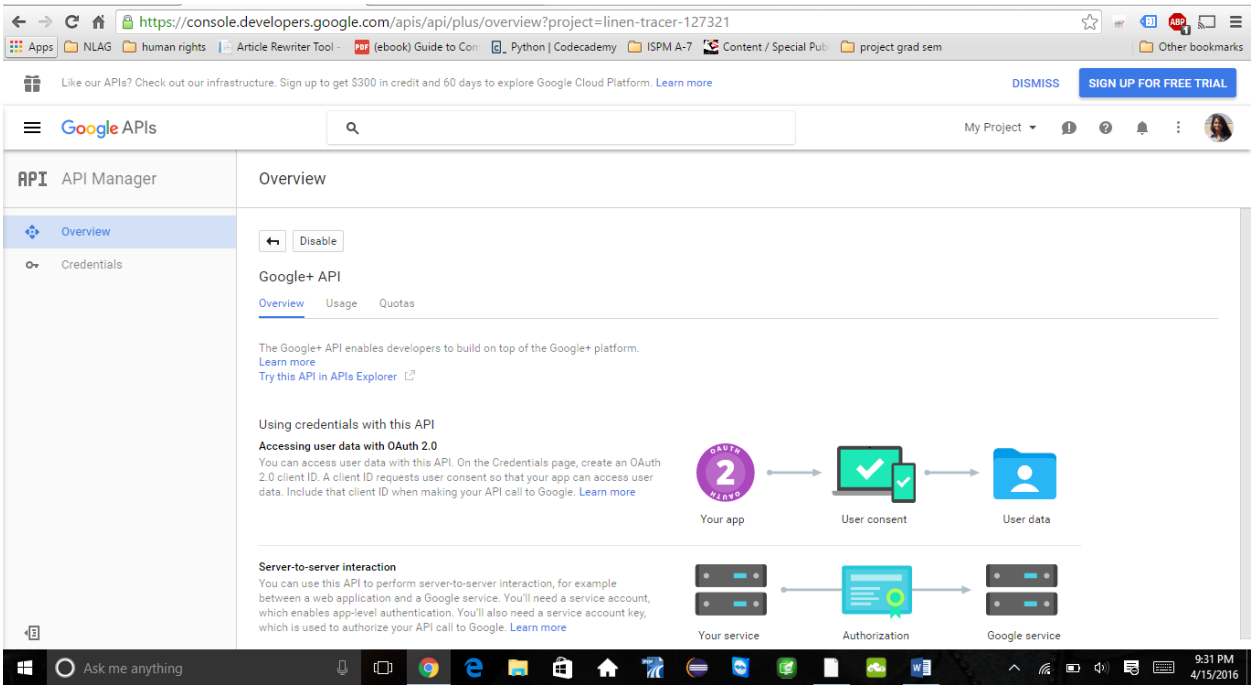

Figure 8: Oauth 2, Google API for Client ID and secret key and authentication.

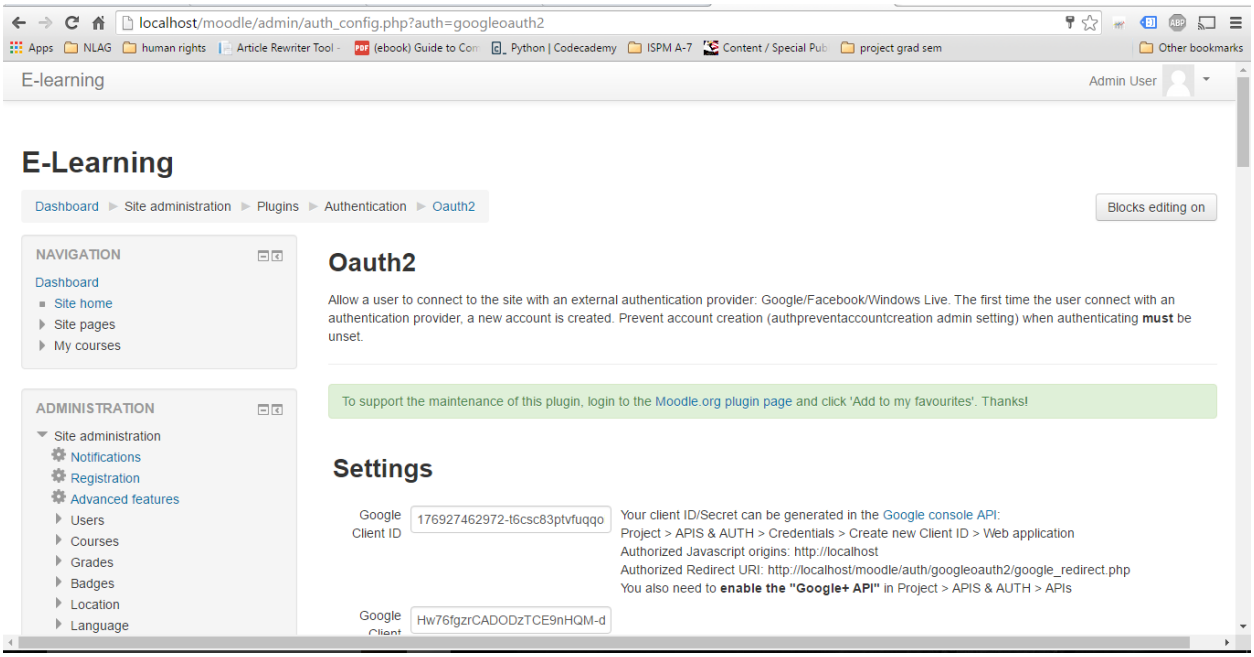

Figure 9: Oauth 2, Pasting Client ID in the settings in the text box beside Client ID

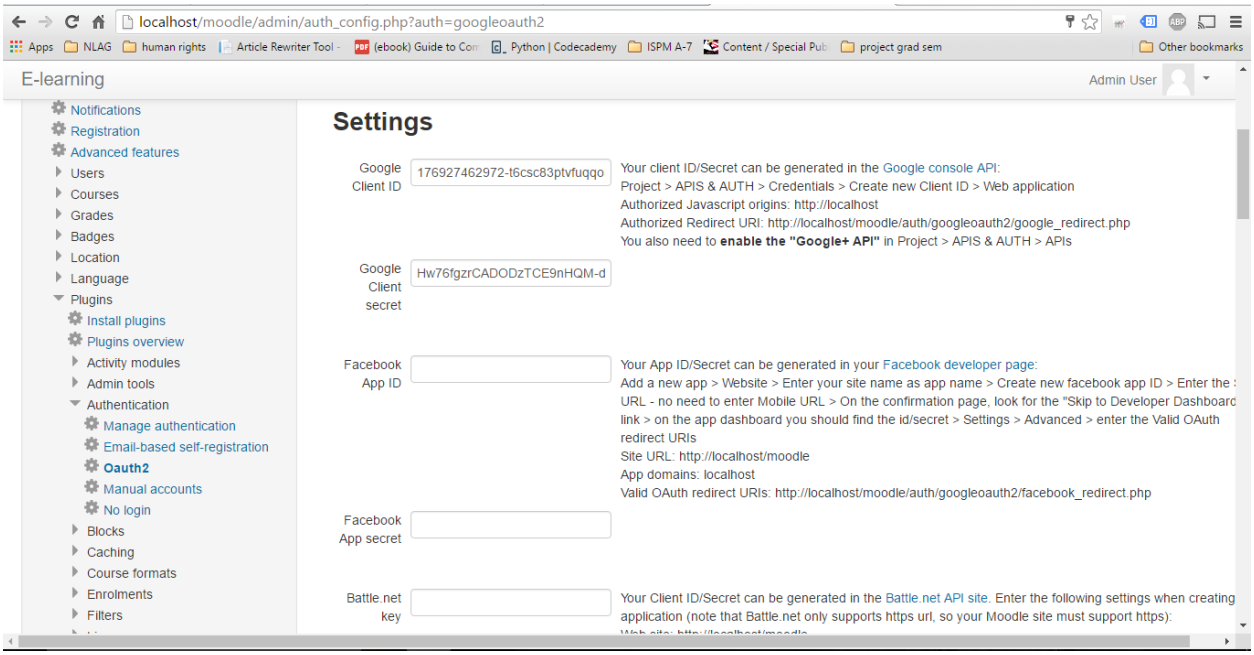

Figure 10: Oauth 2, Pasting Client key in the settings in the text box beside Client key

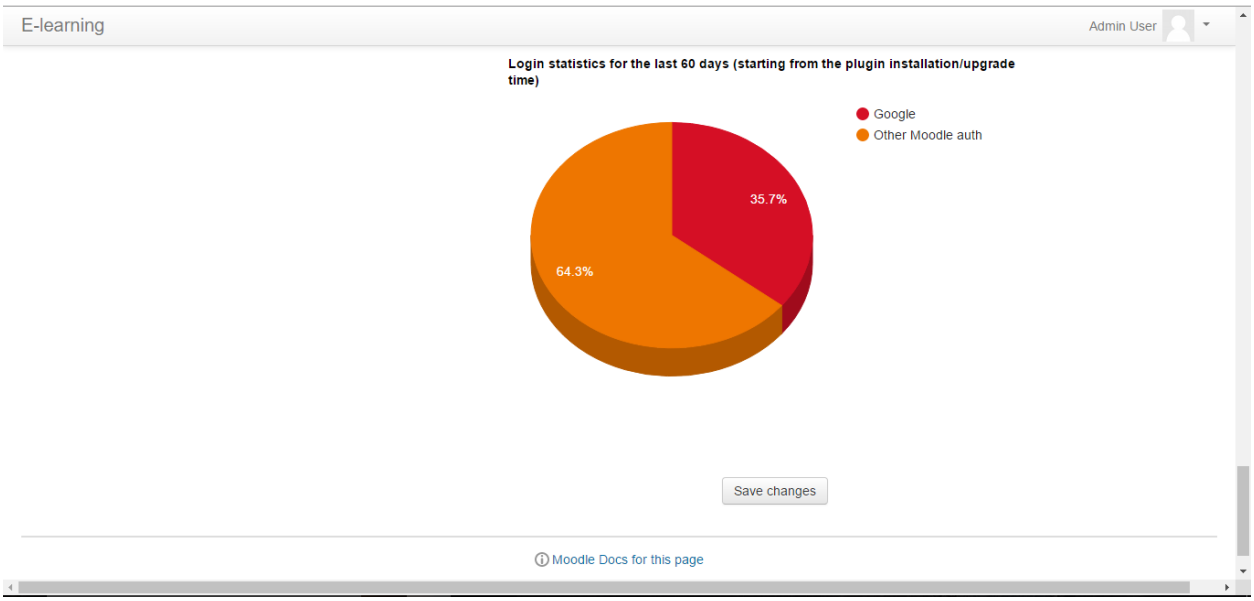

Figure 11: After saving changes we can see the pie chart where it's made 35.7% for Google and rest for Other Moodle authors

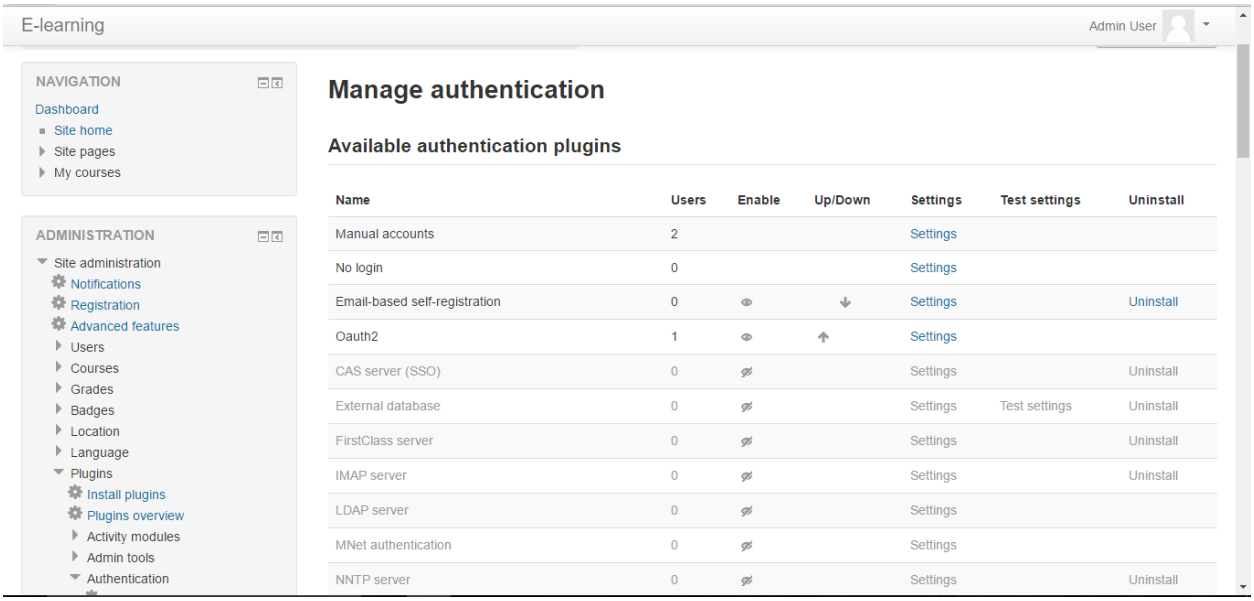

Figure 12: We can see the table for the manual accounts and Oauth 2 account users.

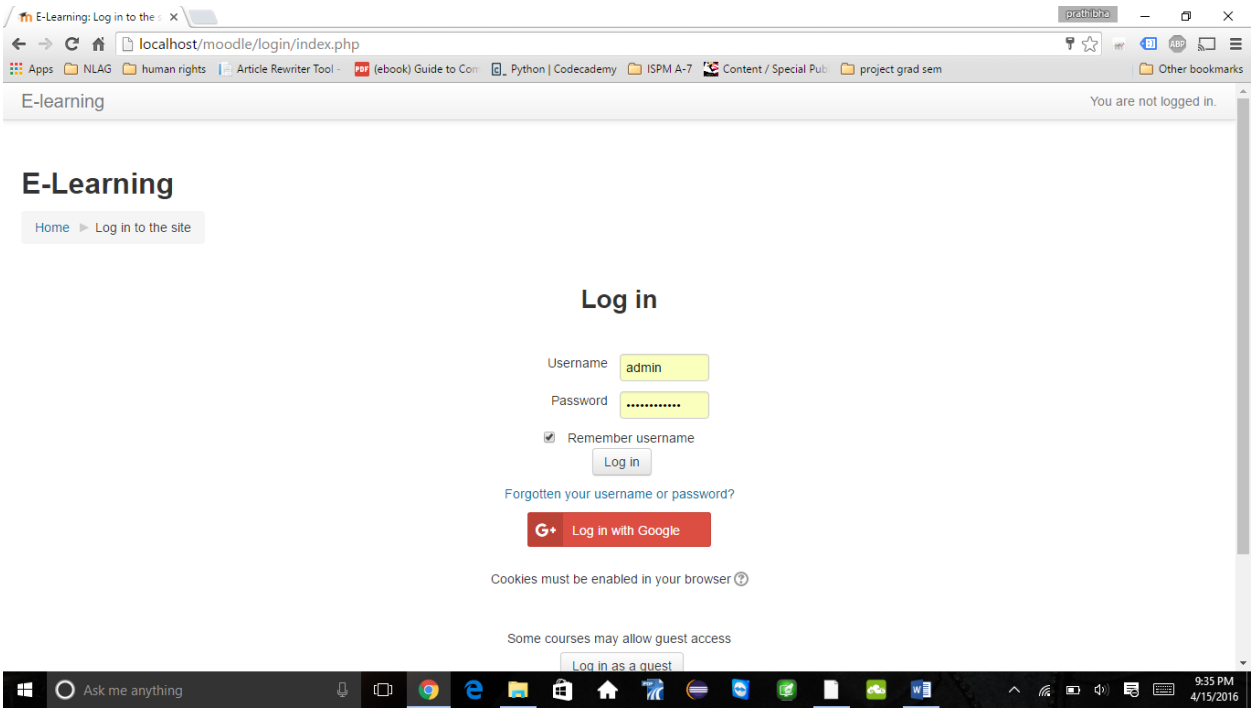

Figure 13: New login page for Moodle after Google easy login (oauth2)

5 Internal/external Interface Impacts and Specification

Replace this section with a description of all internal and/or external interface changes. It should provide sufficient detail to begin updating user documentation, External Interface Specifications, and or Application Interface documents. Also describe here the impacts to any data structures that are shared across design units. If this Project design requires persistent data, describe the persistent data here at a high level.

6 Design Units / Modules

In this project we have only a few simple and limited modules included as Admin, course registration, VPL assignments and grading, enrolling students, students' portal.

6.1 Admin Module

Admin can be the teacher or the course manager. In this project Admin is the manager and the teacher. Duties or the privileges of the admin is that he/she can add or delete course make changed to the course, enroll or un-enroll the students, add the activities like assignments and tests.

6.2 Adding course

Adding a new course, it is pretty simple, login Moodle as admin, turn editing on and then click "Add Course" write the full course name as JAVA Programming Language, Short name as JAVA, in description box "WHAT IS JAVA

Java is a programming language and computing platform first released by Sun Micro-systems in 1995. There are lots of applications and websites that will not work unless you have Java installed, and more are created every day. Java is fast, secure, and reliable. From laptops to data-centers, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!" In Course ID we have given CS123, leave all other options default and save.

Now you can see that JAVA Programming Language course has been created.

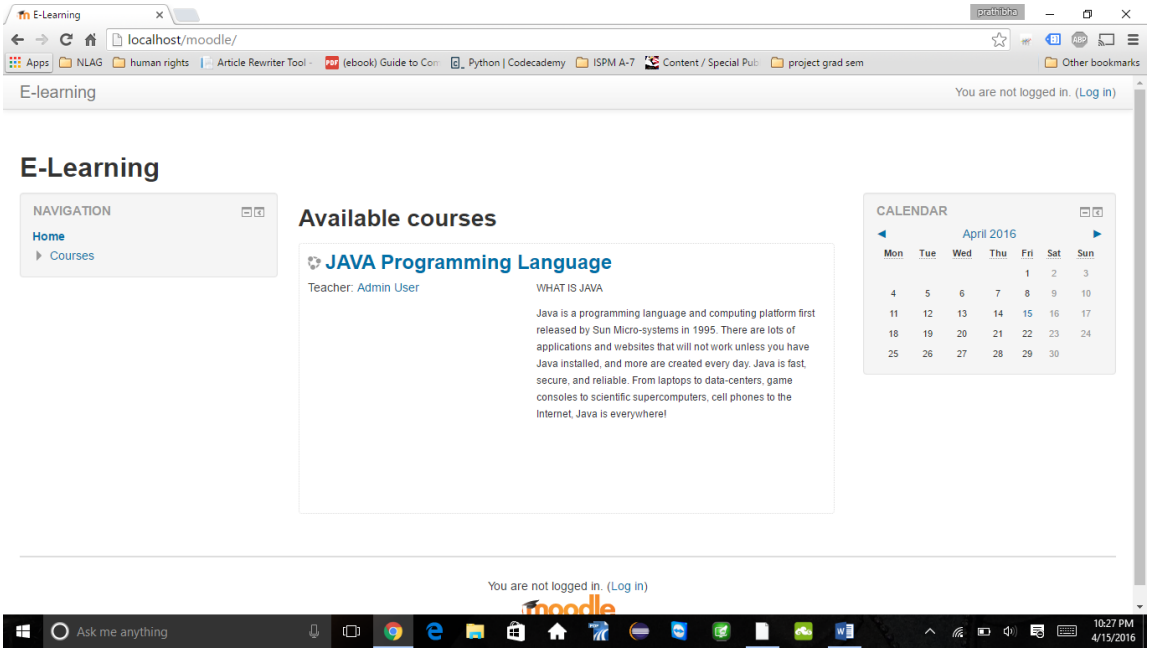

Figure 14: Login Page after adding course

6.3 Enrolling Users

Select the course and under Site Administration turn editing on, select edit settings users enrolled users. Select admin and change role to teacher. In incognito tab in Google chrome create a new account and keep logged in, come back to admin page and refresh the page now you can see new user, select the user enroll it and the role of the user is student. You can enroll how many users as you can.

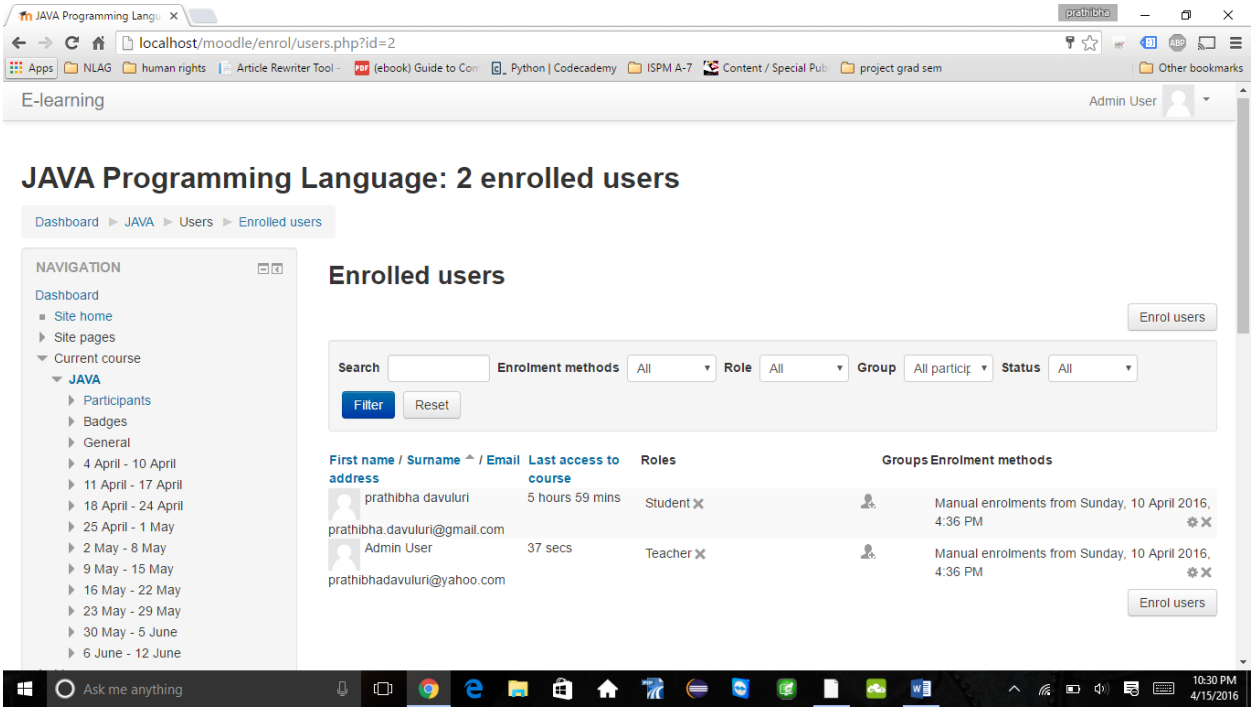

Figure 15: Enrolled users for the course

6.4 Adding and Grading an Assignment

a. Adding a regular text based assignment

Login in Moodle as admin (teacher) now select the course JAVA and turn editing on, there are weekly dates available, select any week and select “add an activity or resource”, you can find many options like quiz, assignment, test, virtual programming lab and so on. To add a simple text assignment, you can select regular assignment radio button and press add write the name of the assignment and write full or short description. In the submission part you can set the submission date and time, maximum files to be submitted and we can enable the text box and enter number of words in maximum to be used and maximum number of files can be uploaded. In grade section you can change grade type or leave it default point, and changes in maximum points or leave them default then click either save and display or save and return to course.

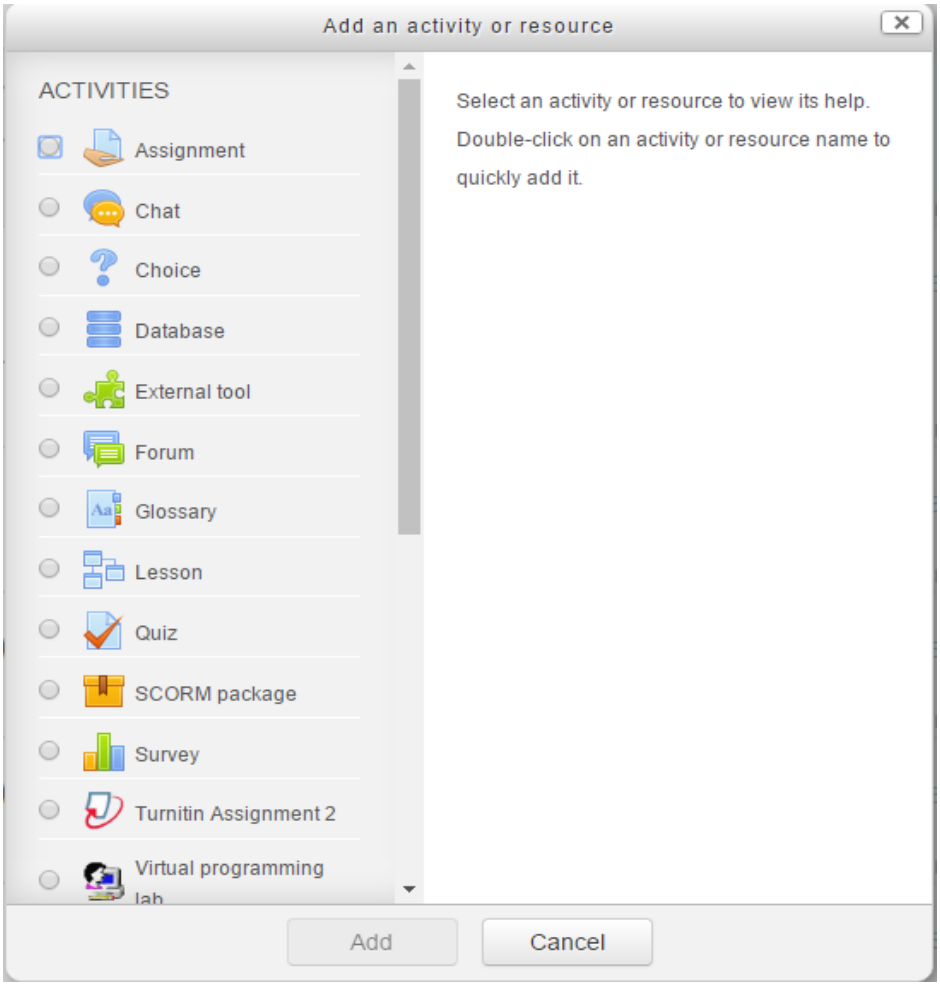

Figure 16: Add activity or resource page

b. Adding a Virtual Programming Lab Assignment

We want to add a programming assignment, so in add an activity or resource select virtual programming lab and click add. We named our assignment “Programming Assignment 1” where in short description “Write a simple java code for Hello World” and in full description “You need to write a Java program that outputs the string "Hello World!"”. In the submission section you can set the submission date and time, maximum files to be submitted and maximum number of files can be uploaded. In grade section you can change grade type or leave it default point, and changes in maximum points or leave them default then click either save and display or save and return to course. Now left panel under “VPL Administration” select “Execution options”. The Execution options panel opens leave the “Based on” default and change the rest options (Run, Debug, Evaluate, evaluate just on submission and Automatic Grade) to “yes” and click save options.

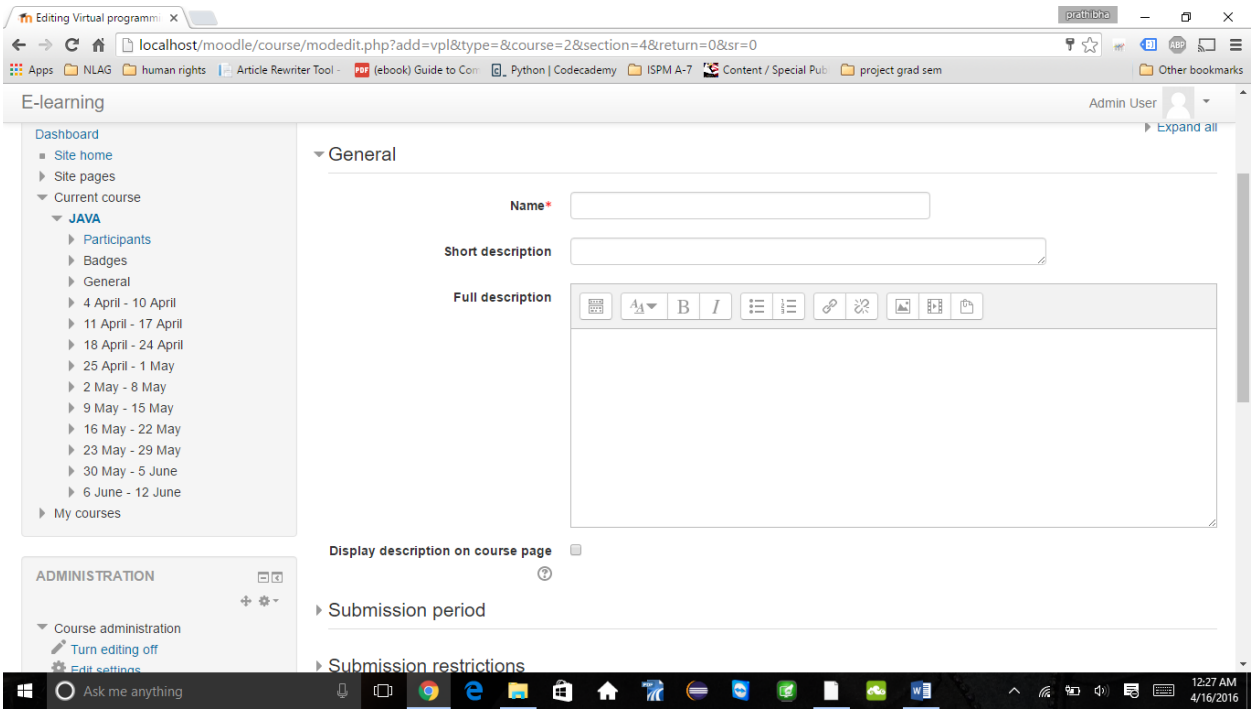

Figure 17: Virtual Programming Lab Assignment Page

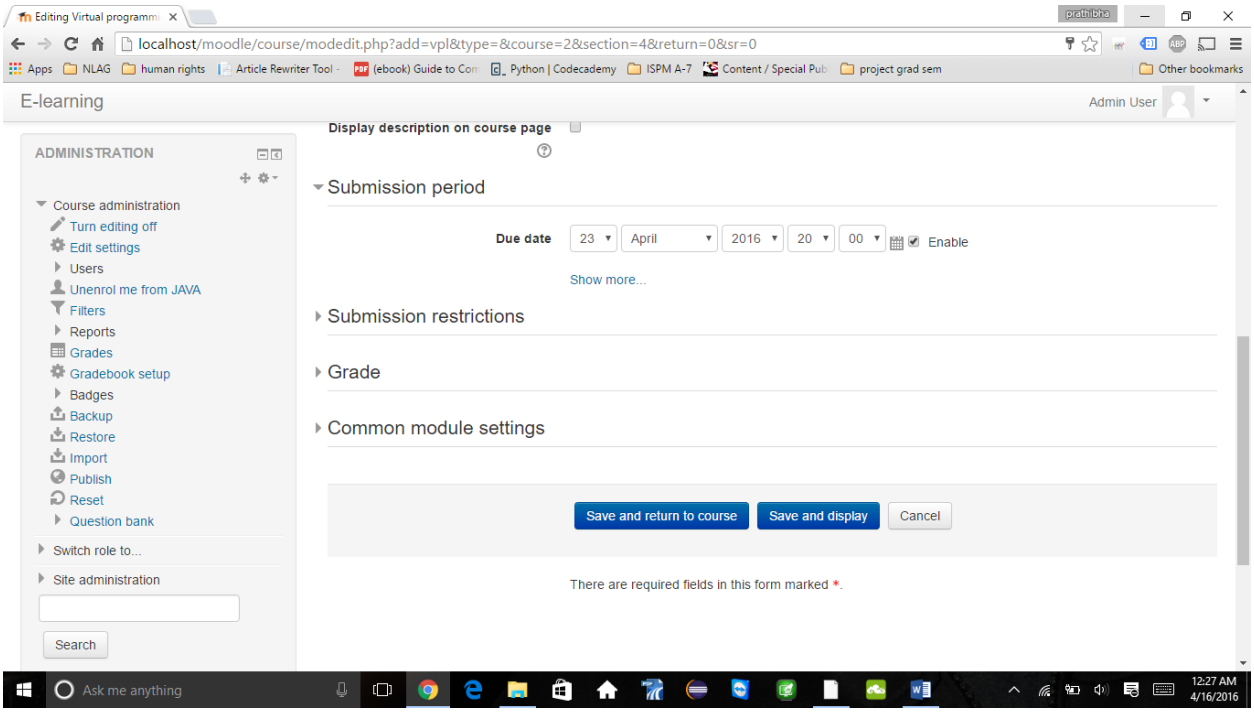

Figure 18: Virtual Programming Lab Assignment Submission and Grade settings

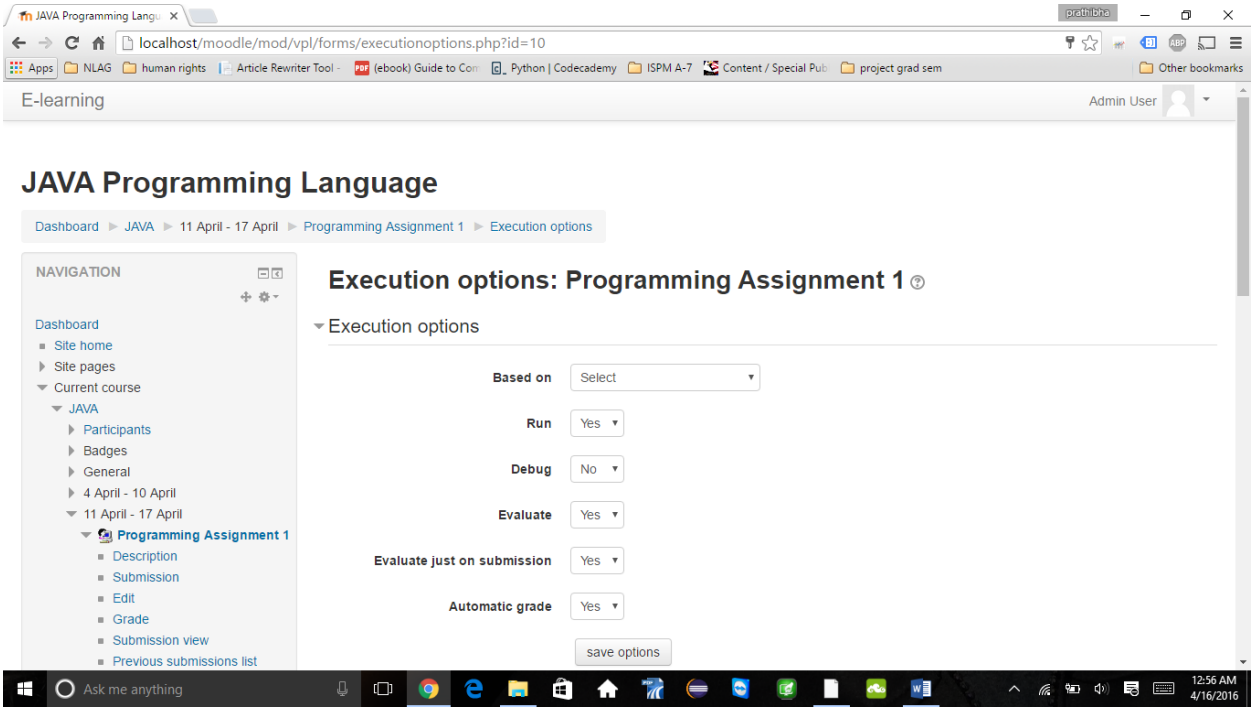

Figure 19: Programming Assignment 1, Execution options page

Requested Files:

Left corner under VPL administration select “Requested Files” create a new file name it “HelloWorld.java” and press create, in that file leave few comments say

// put your code here.

// Make sure that your program does not add extra \n at the end of the output!
and save the file.

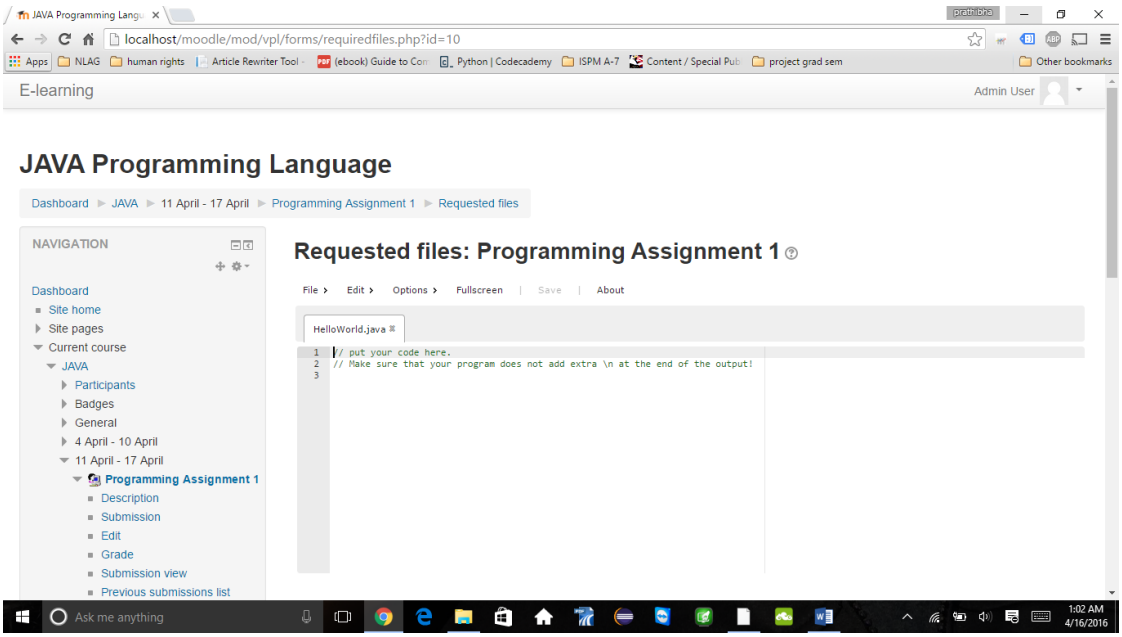

Figure 20: Programming Assignment 1, Requested Files

Execution Files:

Below VPL administration select “Advanced Settings” and then select “Execution files” we find vpl_run.sh, vpl_debug.sh, vpl_evaluate.sh, vpl_evaluate.cases. Place all these codes under those default files.

vpl_run.sh

#! /bin/bash

cat > vpl_execution << 'EOF'

#! /bin/bash

javac -J-Xmx128m HelloWorld.java

java HelloWorld

EOF

chmod +x vpl_execution

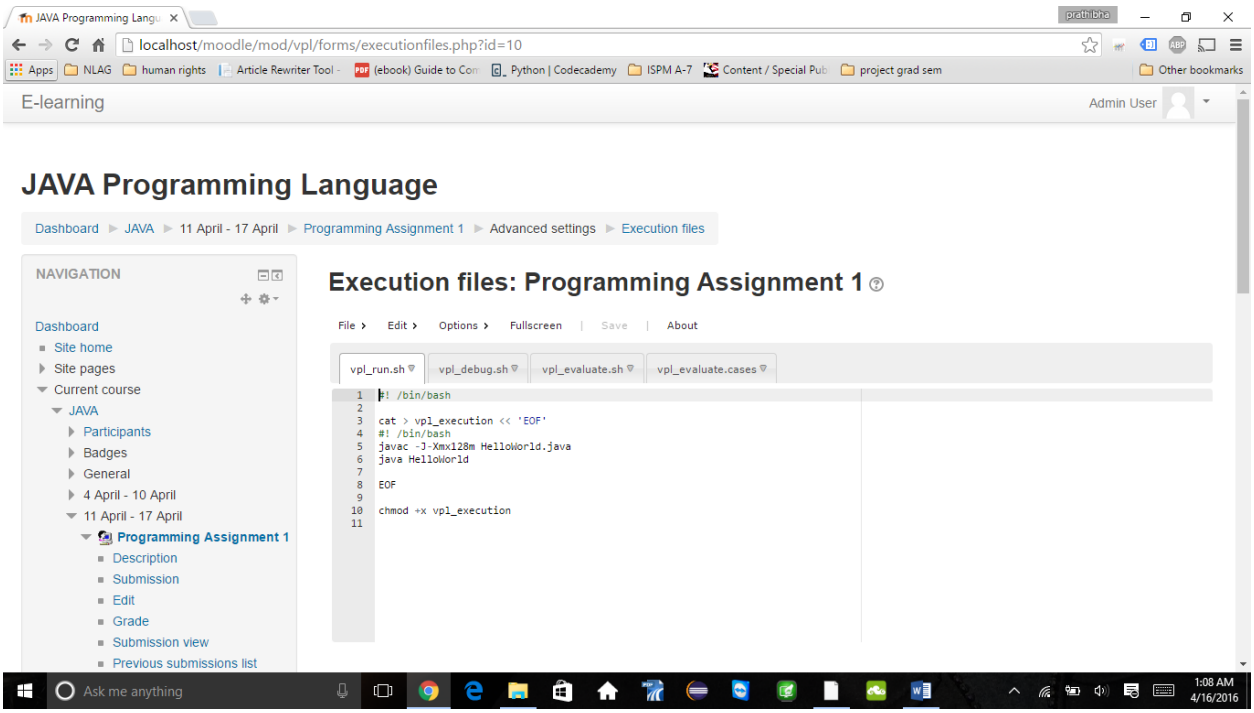

Figure 21: Programming Assignment 1, Execution files, vpl_run.sh

vpl_debug.sh

//leave empty

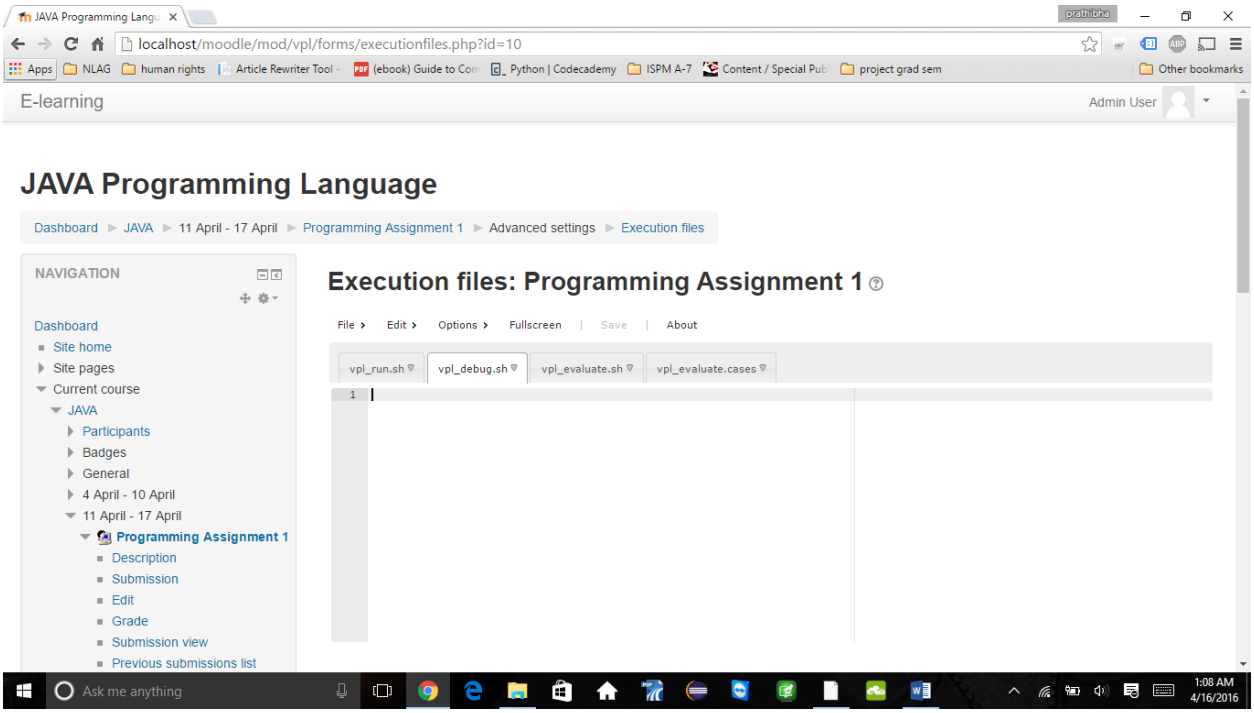

Figure 22: Programming Assignment 1, Execution files, vpl_debug.sh

vpl_evaluate.sh

//leave empty

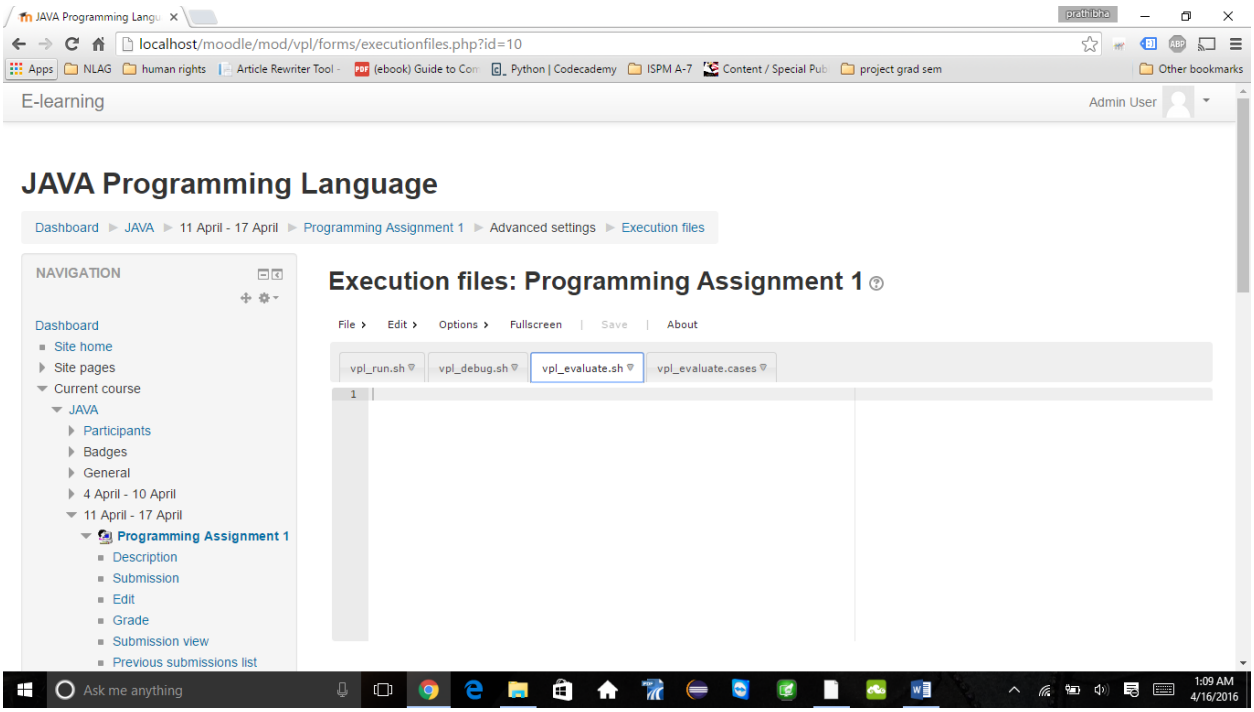

Figure 23: Programming Assignment 1, Execution files, vpl-evaluate.sh

vpl_evaluate.cases

case = Test 1

output = Hello World

case = Test 2

output = Hello, World

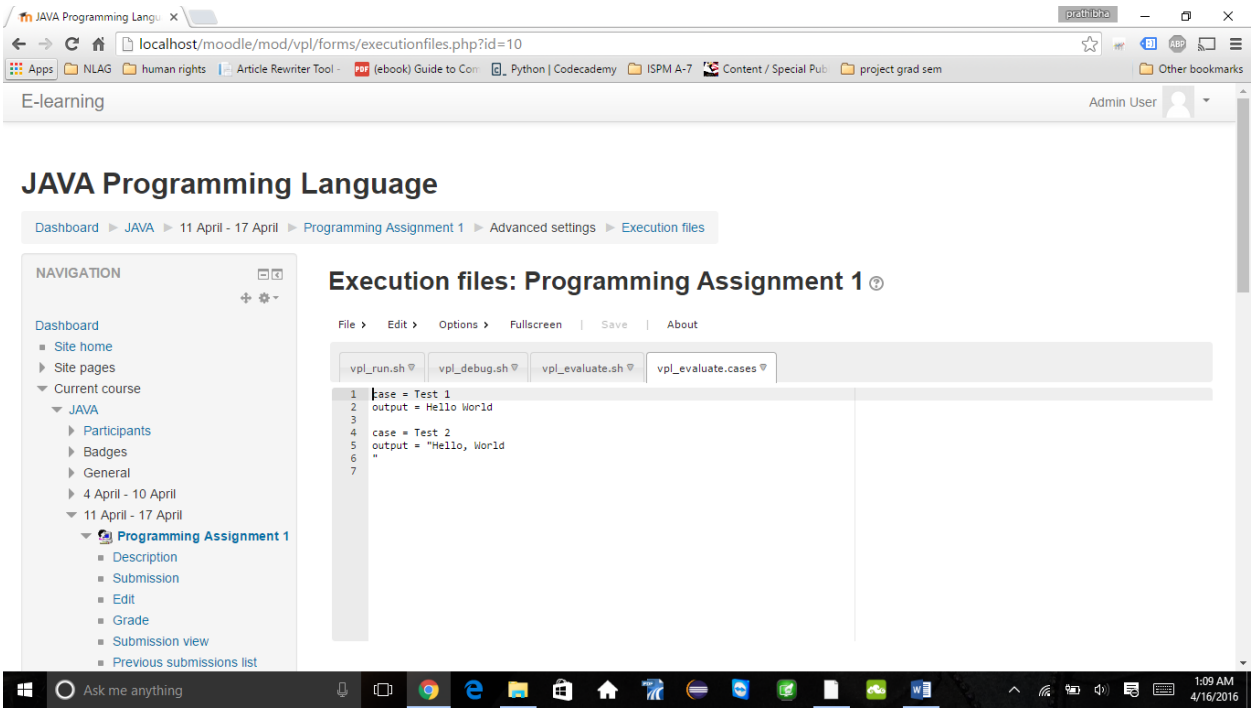

Figure 24: Programming Assignment 1, Execution files, vpl_evaluate.cases

Now save the files. Below VPL administration select “Test activity” (to check whether those file which we have created are working or not, and also to check how well the evaluation is carried out) and then choose “Edit” we can see that HelloWorld.java file is opened now we have to write down the simple java code where for Hello World to print.

```
public class HelloWorld {

   public static void main(String[] args) {

      System.out.println("Hello World");

   }

}
```

Click Save, Run and Evaluate.

When we click Run the Console is opened and Hello World is printed.

When we click Evaluate the process takes few more seconds and the result is shown. Hence the codes and all process is working good.

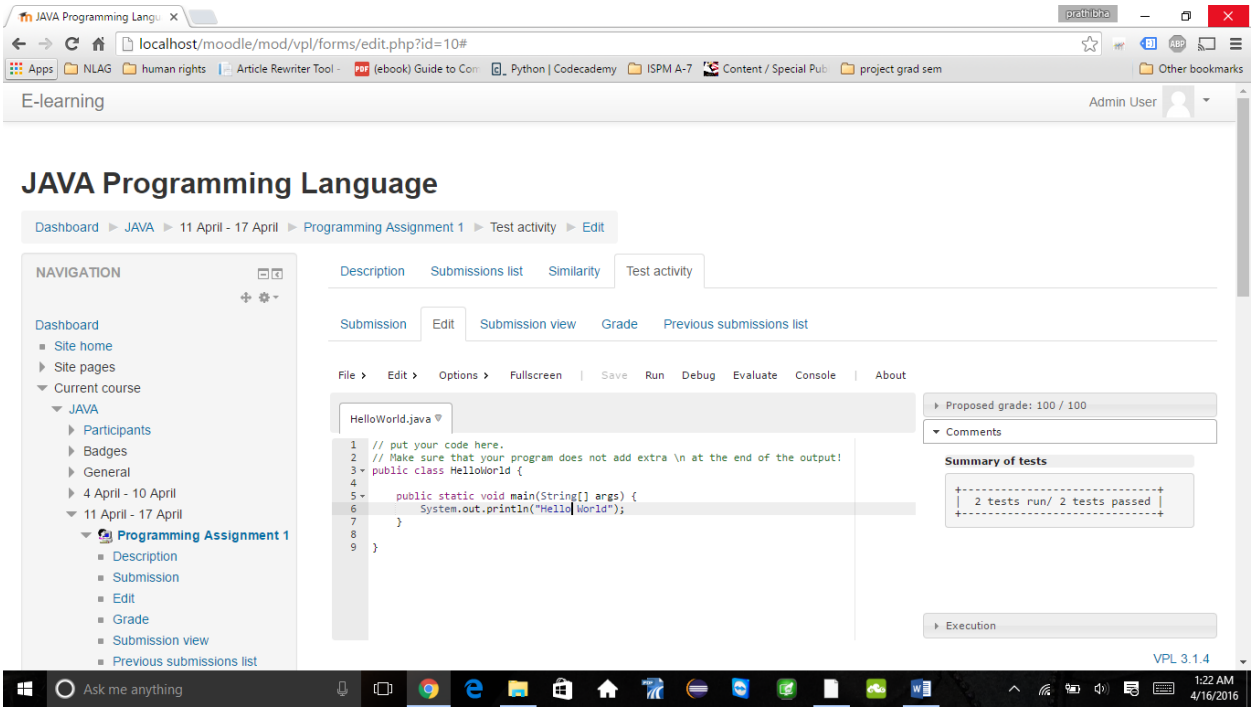

Figure 25: Programming Assignment 1, Test Activity, Grading

6.5 Student Portal

When a student login Moodle under their course they find their Assignment “Programming Assignment 1”, click on that, description and “Requested files” will be seen for them. A student can submit their work file or they can also click edit and the “HelloWorld.java” file will be opened and they can type their code in it. After that they save, run and evaluate their work and they get their work graded. Hence Java Auto Grader.

Here we can give more assignments and more complicated Programming Assignments and check that the work is been Graded automatically. As VPL already has anti plagiarism there is no need to add any extra plagiarism plugins for Moodle.

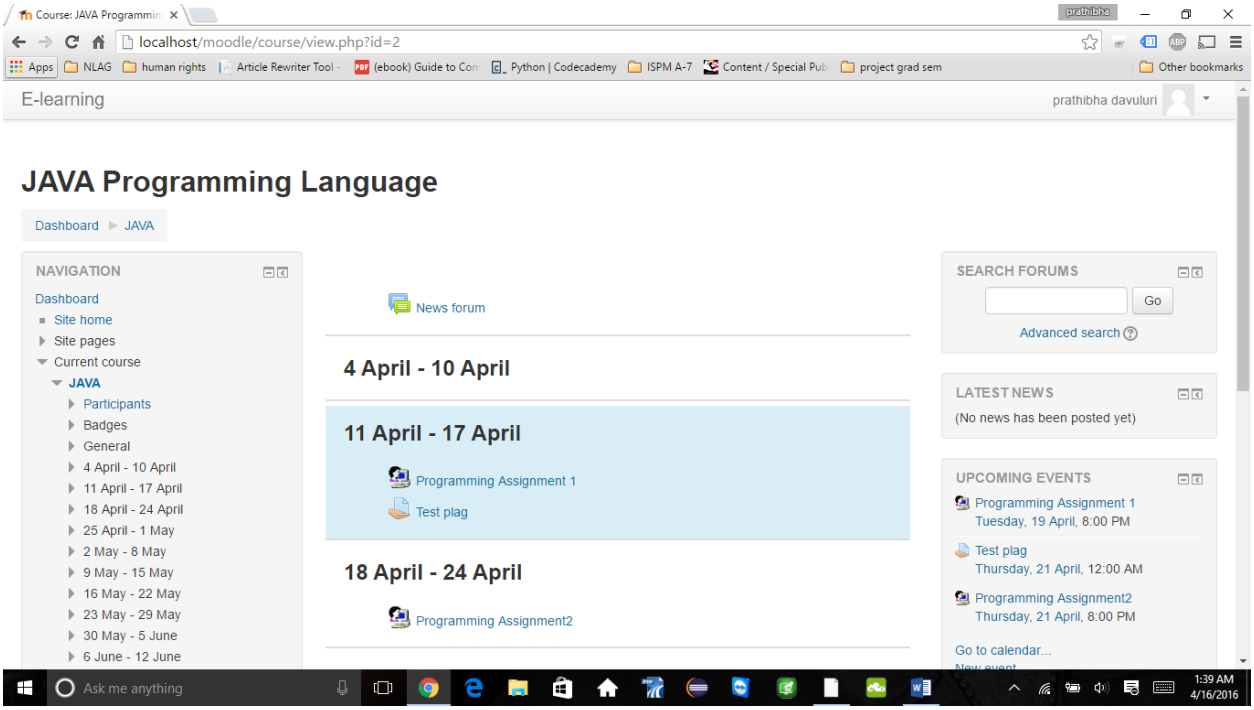

Figure 26: Student E-learning Dashboard, Course page

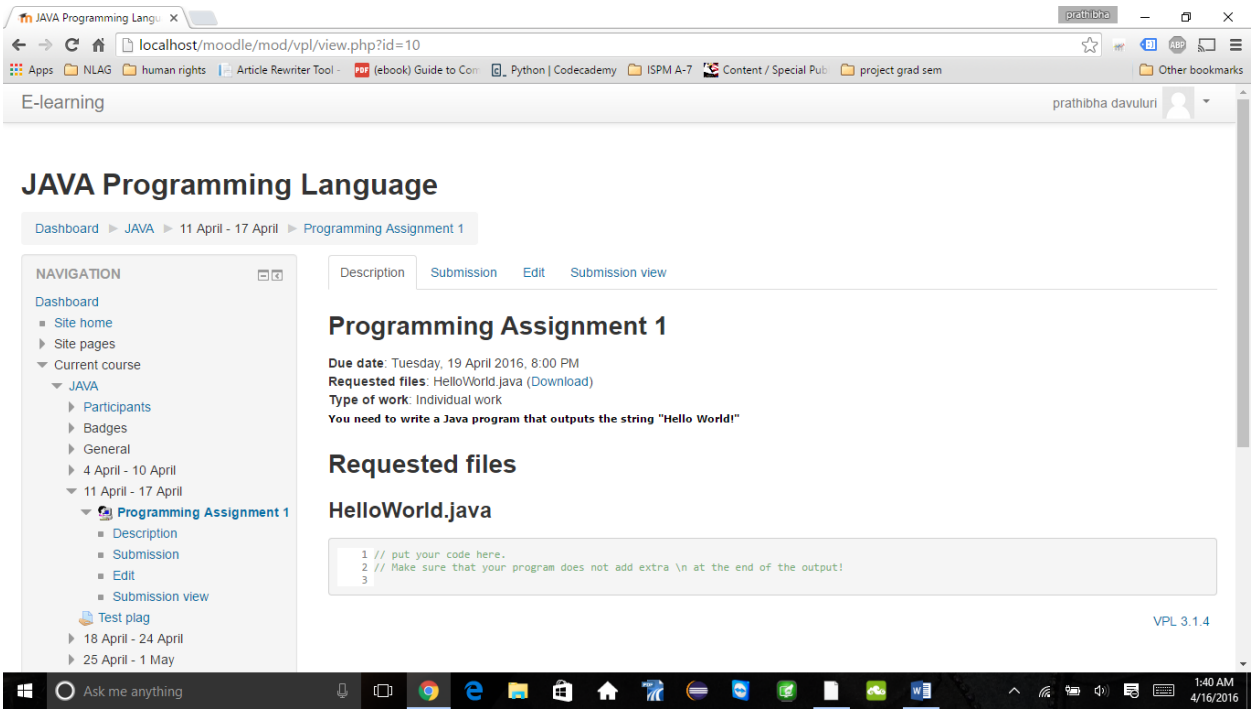

Figure 27: Programming Assignment 1, Description Page

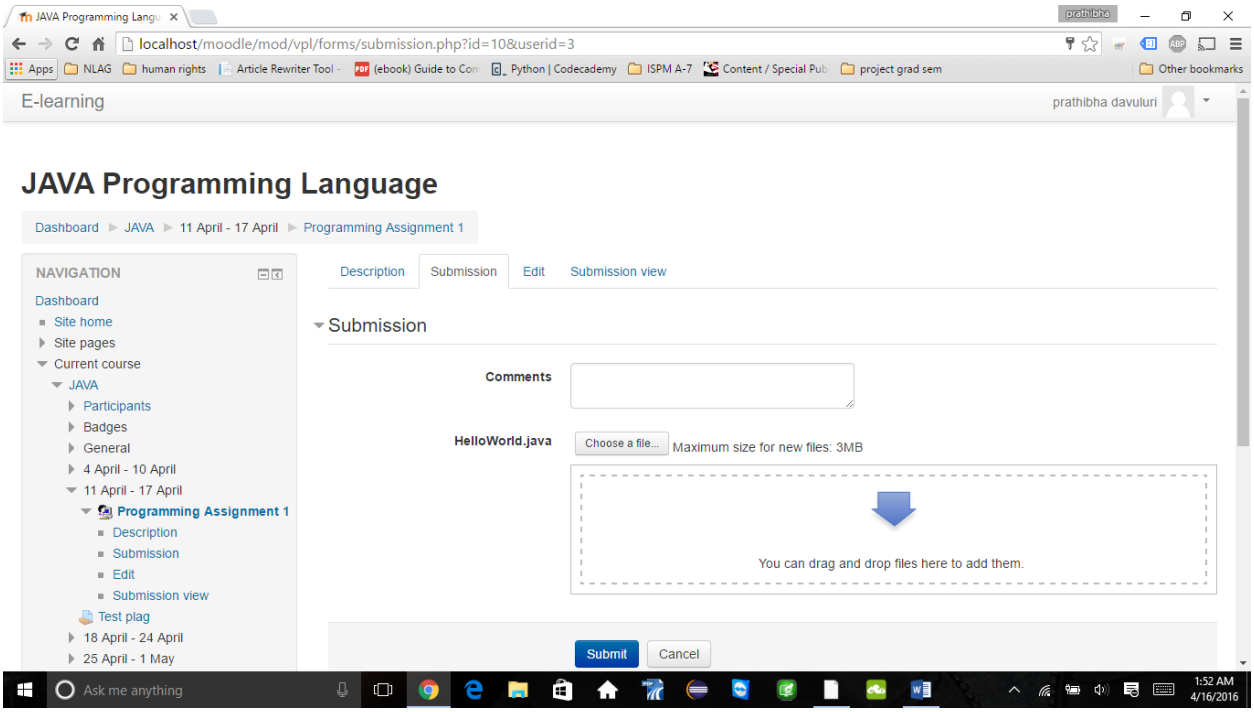

Figure 28: Programming Assignment 1, Submission Page

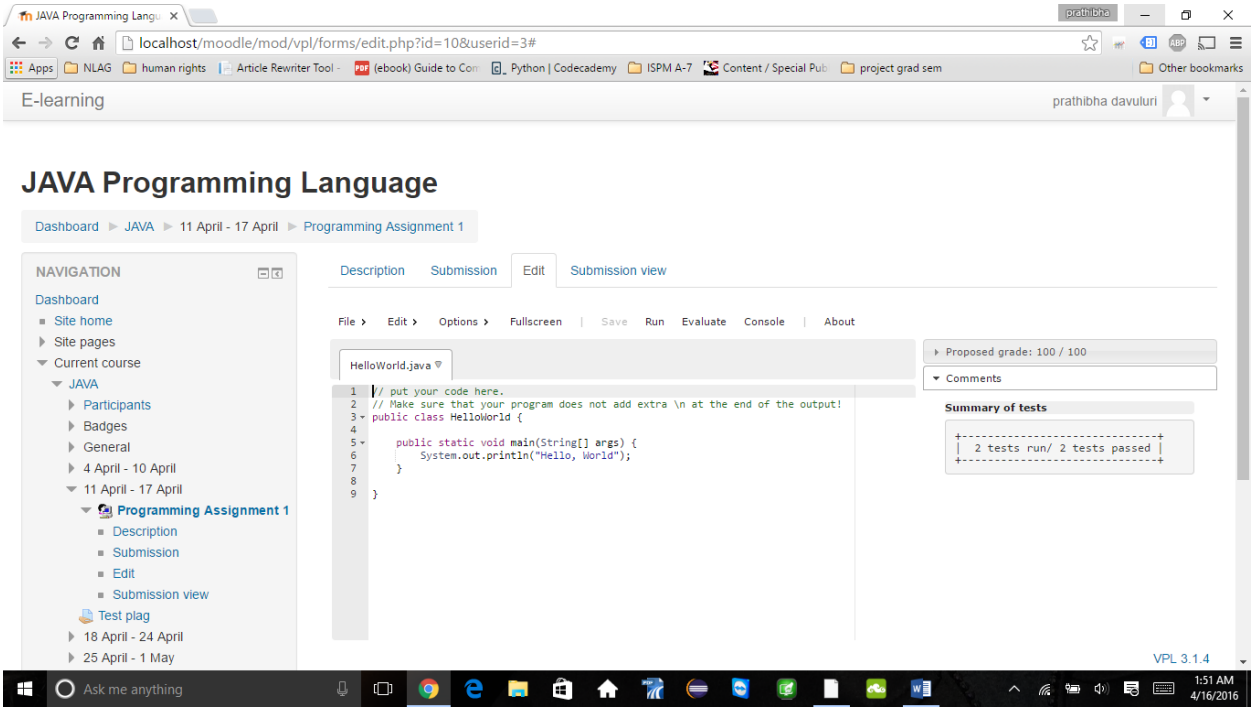

Figure 29: Programming Assignment 1, Edit and Evaluation Page

7 Issues

There was an issue raised by the professor under whom we were making this project we should add an activity VPL assignment where the student has to use scanner and take the input and print out the output as Hello Name. In this assignment we came to know that the student when properly writes the program is getting full score and the student who just writes the simple print statement Hello Name is getting a partial score but not a zero. Later we have built up a compiler code where it catches and reads the print statement and gives a zero for the students who just copy the output shown in the comments box. While adding another class with same name extends the class where the students work is being taken and compiled and grades accordingly.

8 Conclusion

The fast increase we see in CS courses nationwide are two pressures that require arrangements. One option is to adapt the way we teach and evaluate students in programming intensive classes. Our early experience with VPL is positive. The wide array of programming languages VPL supports, its robustness of implementation, and the flexibility it offers compensate for its complexity of use, and its currently sparse documentation. We have started releasing scripts we have generated for various assignments in an effort to share ideas, and solutions, hoping others can benefit from our experience.

9 References

1. *Dominique Theibaut (19 September 2014) Moodle VPL Tutorials. Retrieved from: http://www.science.smith.edu/dftwiki/index.php/Moodle_VPL_Tutorials*
2. *Aldo Von Wangenheim (02 January 2016). Developing Programming Courses with Moodle and VPL. Retrieved from: https://www.researchgate.net/publication/288944522_Developing_Programming_Courses_with_Moodle_and_VPL_-_The_Teacher's_Guide_to_the_Virtual_Programming_Lab*
3. *Dominique Theibaut (2015) Automatic Evaluation of Computer Programs using Moodle's Virtual Programming Lab Module. Retrieved from: https://pdfs.semanticscholar.org/c62c/7c01d6a5ee6ea2b48adc9076d237841c0cea.pdf*